(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 472 812 B1

(12) EUROPÄISCHE PATENTSCHRIFT

(45) Veröffentlichungstag und Bekanntmachung des Hinweises auf die Patenterteilung:
06.05.1998 Patentblatt 1998/19

(51) Int Cl.⁶: **G06F 9/44**, G06F 9/45

(21) Anmeldenummer: **91107079.5**

(22) Anmeldetag: **02.05.1991**

(54) **Verfahren zum Aendern einer in einem Computer eines Gerätes abgespeicherten Maschinensprachenfassung eines ersten Programms in eine Maschinensprachenfassung eines durch mindestens eine Aenderung vom ersten Programm abgeleiteten zweiten Programms**

Method to change an object code version of a first program stored in the computer of an appliance into an object code version of a second program which was derived by at least one change to the first program

Méthode pour changer une version du code objet d'un premier programme mémorisé dans le calculateur d'un dispositif en une version du code objet d'un deuxième programme qui a été dérivé par au moins un changement du premier programme

(84) Benannte Vertragsstaaten:
CH DE GB LI SE

(30) Priorität: **28.08.1990 CH 2793/90**

(43) Veröffentlichungstag der Anmeldung:
04.03.1992 Patentblatt 1992/10

(73) Patentinhaber: **Landis & Gyr Technology Innovation AG**
6301 Zug (CH)

(72) Erfinder:
- **Wehrli, Herbert**
  CH-8865 Bilten (CH)
- **Meyer, Mark**
  CH-6330 Cham (CH)

(56) Entgegenhaltungen:
EP-A- 194 822
EP-A- 323 707
WO-A-83/01847
US-A- 4 558 413

- **SOFTWARE-PRACTICE AND EXPERIENCE, Band 17, Nr. 7, Juli 1987, Seiten 455-467, Chichester, GB; M.K. CROWE: "Dynamic compilation in the UNIX environment"**

Printed by Jouve, 75001 PARIS (FR)

EP 0 472 812 B1

einen grösseren Speicherbedarf benötigt als in derjenigen des ersten Programms (Y_1),

- unter Freigabe des bisher belegten Speicherbereichs zugunsten des Inhaltes eines oder mehrerer anderer Segmente, in einem freien Speicherbereich des Computers (8) der Zentraleinheit (6) speichert sowie

- Referenzangaben, die sich in anderen Segmenten auf das betreffende Segment beziehen, korrigiert und auf den neuesten Stand bringt,

- dass dem "Linker"-Programm (20) ein Komparator/Generator-Programm (21) nachgeordnet ist, welches

  - alle Bytes der beiden Maschinensprachenfassungen (X_1, X_2) miteinander vergleicht und

  - ein Unterschiedlichkeiten-Programm (δX) erzeugt, welches nur mehr die für die beiden Maschinensprachenfassungen (X_1, X_2) unterschiedlichen Bytes mit ihnen zugeordneten Adressen (a1, a2, a3, a4) enthält, und

- dass das Unterschiedlichkeiten-Programm (δX) dem Computer (8) des Gerätes (1, 2, 3, 4 oder 5) zugeleitet wird und seine Bytes dort unter ihren zugeordneten Adressen (a1, a2, a3, a4) abgespeichert werden, wobei sie zusammen mit den bereits vorhandenen, nicht geänderten Bytes der Maschinensprachenfassung (X_1) des ersten Programms (Y_1) des Gerätes (1, 2, 3, 4 oder 5) dessen Maschinensprachenfassung (X_2) des zweiten Programms (Y_2) bilden.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass das "Compiler"-Programm (17) jedem der Segmente einen eindeutigen Segmentnamen zuordnet, der für beide Programme (Y_1, Y_2) identisch ist.

3. Verfahren nach Anspruch 1 oder 2, dadurch gekennzeichnet, dass die Segmentinformationen jeweils mindestens einen Segmentnamen, eine Startadresse des Segmentes im physikalischen Speicher, einen Speicherbedarf des Segmentes und eine maximal mögliche Grösse des Segmentes beinhalten.

4. Verfahren nach einem der Ansprüche 1 bis 3, dadurch gekennzeichnet, dass das "Compiler"-Programm (17) alle Segmente des anstehenden Programms (Y_1 bzw. Y_2) in Sektionen zusammenfasst und dass das "Linker"-Programm (20) im Computer (8) der Zentraleinheit (6) die Sektionen anhand einer Spezifikation in einem Speicherbereich plaziert.

5. Verfahren nach Anspruch 4, dadurch gekennzeichnet, dass die Segmentinformationen jeweils eine Sektionszugehörigkeit des Segments beinhalten.

Claims

1. A method of changing a machine language version (X_1) of a first program (Y_1) which is stored in a computer (8) of an apparatus (1, 2, 3, 4 or 5) into a machine language version (X_2) of a second program of a second program (Y_2) which is derived from the first program (Y_1) by at least one change, wherein at respective different moments in time associated first and second source versions (Q_1 and Q_2 respectively) of the two programs (Y_1, Y_2) are converted in a computer (8) of a central unit (6) by means of a compiler/linker-program (19) into the associated first and second machine language versions (X_1 and X_2 respectively),
   characterised in that

   - at the respective moment in time of conversion of a source version (Q_1 and Q_2) into the associated machine language version (X_1 and X_2) in the computer (8) of the central unit (8)

     - a compiler-program (17) of the compiler/linker-program (19) divides the source version (Q_1 and Q_2 respectively) in question into segments which have a direct relationship with the content of the source version (Q_1 and Q_2 respectively) and to which there corresponds a respective segment in the associated machine language version (X_1 and X_2 respectively), the segments in the subsequent compiler passes each forming a respective undividable unit, and
     - a linker-program (20) of the compiler/linker-program (19) stores segment information in respect of the segments of the program (Y_1 and Y_2 respectively) in question in an intermediate data file (Z_1 and Z_2

respectively), and

- the linker-program (20) during conversion of the source version (Q_2) of the second program (Y_2) into its machine language version (X_2)

  - reads the segment information of the first program (Y_1) stored in a first intermediate data file (Z_1) and compares same to the segment information of the second program (Y_2) stored in a second intermediate data file (Z_2),
  - stores the content of each segment which in the machine language version (X_2) of the second program (Y_2) requires at most the same storage demand as in that of the first program (Y_1), in both machine language versions (X_1, X_2) at a respectively identical address in the computer (8) of the central unit (6), and
  - stores the content of each segment which in the machine language version (X_2) of the second program (Y_2) requires a greater storage demand than in that of the first program (Y_1),

    - freeing the previously occupied storage region in favour of the content of one or more other segments, in a free storage region of the computer (8) of the central unit (6), and
    - corrects reference specifications relating in other segments to the segment in question and sets them to the newest status,

- arranged downstream of the linker-program (20) is a comparator/generator-program (21) which

  - compares together all bytes of the two machine language versions (X_1, X_2), and
  - generates a differences program (δX) which only contains the bytes which are different for the two machine language versions (X_1, X_2), with addresses (a1, a2, a3, a4) associated therewith, and

- the differences program (δX) is passed to the computer (8) of the apparatus (1, 2, 3, 4 or 5) and its bytes are stored there at their associated addresses (a1, a2, a3, a4), wherein they together with the already present, unchanged bytes of the machine language version (X_1) of the first program (Y_1) of the apparatus (1, 2, 3, 4 or 5) form its machine language version (X_2) of the second program (Y_2).

2. A method according to claim 1 characterised in that the compiler-program (17) associates with each of the segments a definite segment name which is identical for both programs (Y_1, Y_2).

3. A method according to claim 1 or claim 2 characterised in that the items of segment information each contain at least a segment name, a start address of the segment in the physical store, a storage requirement of the segment, and a maximum possible size of the segment.

4. A method according to one of claims 1 to 3 characterised in that the compiler-program (17) assembles all segments of the program (Y_1 and Y_2 respectively) to be dealt with into sections and that in the computer (8) of the central unit (6) the linker-program (20) places the sections by means of a specification in a storage region.

5. A method according to claim 4 characterised in that the items of segment information each contain a section association of the segment.

Revendications

1. Procédé pour transformer un libellé (X_1) en langage machine, mémorisé dans un ordinateur (8) d'un appareil (1, 2, 3, 4 ou 5), d'un premier programme (Y_1) en un libellé (X_2) en langage machine d'un second programme (Y_2) dérivé au moyen d'une modification du premier programme (Y_1), un premier ou second libellé d'origine associé (Q_1 ou Q_2) des deux programmes (Y_1, Y_2) étant converti, à des instants respectivement différents, dans un ordinateur (8) d'une unité centrale (6), à l'aide d'un programme "Compiler/Linker" (19) en le premier ou second libellé associé (X_1 ou X_2) en langage machine, caractérisé en ce

   - qu'à l'instant de la conversion d'un libellé d'origine (Q_1 ou Q_2) en le libellé associé (X_1, X_2) en langage machine dans l'ordinateur (8) de l'unité centrale (6)

     - un programme "Compiler" (17) du programme "Compiler/Linker" divise le libellé d'origine considéré (Q_1,

Europäisches Patentamt

EUROPÄISCHER RECHERCHENBERICHT

Nummer der Anmeldung

EP 91 10 7079

EINSCHLÄGIGE DOKUMENTE

Kategorie	Kennzeichnung des Dokuments mit Angabe, soweit erforderlich, der maßgeblichen Teile	Betrifft Anspruch	KLASSIFIKATION DER ANMELDUNG (Int. Cl.5)
A	EP-A-0 323 707 (WESTINGHOUSE ELECTRIC CORP.) * Zusammenfassung; Figuren 1A,1B; Seite 3, Zeile 1 - Seite 5, Zeile 4 *	1	G 06 F 9/45 G 06 F 9/445
A	EP-A-0 194 822 (SONY CORP.) * Zusammenfassung; Seite 2, Zeilen 15-28; Seite 3, Zeilen 14-23; Seite 4, Zeilen 5-22; Figur 2 *	1	
A	US-A-4 558 413 (SCHMIDT et al.) * Spalte 9, Zeilen 28-59; Spalte 10, Zeilen 4-3 *	1	
A	WO-A-8 301 847 (WESTERN ELECTRIC CO.) * Zusammenfassung; Figuren 1,2; Seite 3, Zeilen 27-38; Seite 10, Zeilen 30-36 *	1	
A	SOFTWARE-PRACTICE AND EXPERIENCE, Band 17, Nr. 7, Juli 1987, Seiten 455-467, Chichester, GB; M.K. CROWE: "Dynamic compilation in the UNIX environment" * Seite 455, Zeile 1 - Seite 456, Absatz 1; Seite 456, letzter Absatz; Seite 457, letzter Absatz - Seite 459, Absatz 1 *	1,4	
			RECHERCHIERTE SACHGEBIETE (Int. Cl.5)
			G 06 F

Der vorliegende Recherchenbericht wurde für alle Patentansprüche erstellt

Recherchenort	Abschlußdatum der Recherche	Prüfer
Den Haag	26 Juli 91	FONDERSON A.I.

KATEGORIE DER GENANNTEN DOKUMENTE

X: von besonderer Bedeutung allein betrachtet
Y: von besonderer Bedeutung in Verbindung mit einer anderen Veröffentlichung derselben Kategorie
A: technologischer Hintergrund
O: nichtschriftliche Offenbarung
P: Zwischenliteratur
T: der Erfindung zugrunde liegende Theorien oder Grundsätze

E: älteres Patentdokument, das jedoch erst am oder nach dem Anmeldedatum veröffentlicht worden ist
D: in der Anmeldung angeführtes Dokument
L: aus anderen Gründen angeführtes Dokument

&: Mitglied der gleichen Patentfamilie, übereinstimmendes Dokument